Energy Based and Variational Methods in Neural Networks

Guide: Vineeth N Balasubramanian

Sahil Manocha

Department of Computer Science
Indian Institute of Technology Hyderabad

Honors-I Presentation

Outline

Outline

Introduction

Neural Networks

Feed Forward Neural Networks

- Function approximators
  Given
  $X = \{x_1, x_2, \cdots x_n\}$ and
  $y = \{y_1, y_2, \ldots, y_n\}$ find
  $f : X \rightarrow y, f(x_i) = y_i$
- Training via gradient descent (backpropagation)

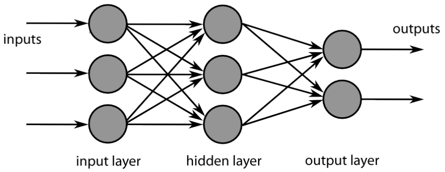


Figure: Image credits [3]

Outline

Introduction

Neural Networks

Boltzmann Machines

- Associative NN like Hopfield Nets
- Stochastic Units
- Intractable probability density function

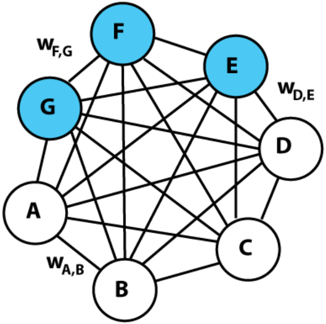


Figure: Image credits [4]

Introduction

Neural Networks

Restricted Boltzmann Machines

- Tractable conditional probability distribution
- Contrastive divergence training procedure

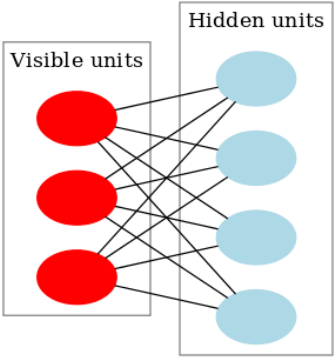


Figure: Image credits [5]

Outline

RBM

Energy

An energy function is defined in an RBM that is minimized in the training of the neural network:

$$E(\mathbf{v}, \mathbf{h}) = -(\mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{h} + \mathbf{h}^T W \mathbf{v})$$

where the probability of a particular training example v is

$$P(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{Z}, \qquad where\ Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$$

Outline

Restricted Boltzmann Machine

Inference

- Marginal is intractable due to normalization
- Conditional probability simplifies

$$P(\mathbf{h}|\mathbf{v}) = \prod_{h_i} P(h_i|\mathbf{v}), \qquad P(\mathbf{v}|\mathbf{h}) = \prod_{v_i} P(v_i|\mathbf{h})$$

$$P(h_i = 1|\mathbf{v}) = \sigma(c_i + \sum_j W_{ij} v_j)$$

$$P(v_j = 1|\mathbf{h}) = \sigma(b_j + \sum_i W_{ij} h_i)$$

Outline

Restricted Boltzmann Machine

Training

The objective of training an RBM is to maximize the log likelihood over the training set. A stochastic gradient descent is performed to reduce the negative log likelihood over the training example. The gradient of the log likelihood is

$$\begin{aligned}\frac{\partial \ln P(\mathbf{v}^{(k)})}{\partial \theta} &= -E_{p(\mathbf{h}|\mathbf{v}^{(k)})}[\frac{\partial E(\mathbf{v}^{(k)},\mathbf{h})}{\partial \theta}] \\ &+ E_{p(\mathbf{h},\mathbf{v})}[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta}]\end{aligned}$$

Restricted Boltzmann Machine

Training

The Contrastive Divergence method approximates the expectation of the gradient as the gradient at *fantasy particle* v'. This fantasy particle is obtained after some predetermined number of K gibbs sampling steps.

$$E_{p(\mathbf{h},\mathbf{v})}[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta}] \approx \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \theta}$$

There are several problems with the training procedure:

- Point estimate not an accurate approximation of expectation
  Remedy : Persistent Contrastive Divergence
- Let p_{data} be the source distribution. p_{model} has high probability at x where $p_{data}(x) \approx 0$.
  Remedy : Diss-CD

Outline

Diss-CD

Introduction

Idea: Initialization of gibbs-chain using "dissimilar" point

What is **dissimilar**?

Any data point not arising out of the source distribution is treated as dissimilar. For MNIST digits, this can be a triangle image.

Outline

Diss-CD

Algorithm

Algorithm 1 Dissimilar Contrastive Divergence Algorithm

Input: RBM(W, b, c), Training data S, Dissimilar data $\bar{S}$, Number of Gibbs cycles K, Number of hidden units n, Number of visible units m
Output: DissCD trained RBM
Initialize $W \sim \left[-\frac{\sqrt{6}}{\sqrt{n+m}}, \frac{\sqrt{6}}{\sqrt{n+m}}\right]$, $b = 0$, $c = 0$
for all $\mathbf{pos_v} \in S, \mathbf{neg_v} \in \bar{S}$ **do**
    $V^{(0)} \leftarrow \mathbf{pos_v}$, $V \leftarrow \mathbf{neg_v}$
    $H \leftarrow \textsc{SampleHGivenV}(V)$
    for $j = 1$ **to** K **do**
        $V \leftarrow \textsc{SampleVGivenH}(H)$
        $H \leftarrow \textsc{SampleHGivenV}(V)$
    end for
    $V' \leftarrow V$
    $w_{ij} = w_{ij} + p(h_i = 1|V^{(0)})v_j^{(0)} - p(h_i = 1|V')v_j'$
    $b_j = b_j + v_j^{(0)} - v_j'$
    $c_i = c_i + p(h_i = 1|V^{(0)}) - p(h_i = 1|V')$
end for

Outline

Diss-CD

Empirical Justification

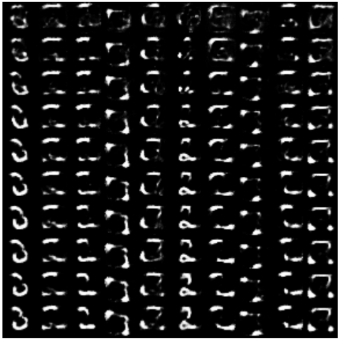


Figure: Visible layer initialized to 10 examples from triangle dataset. Sampled after every 2 Gibbs cycles on **PCD** trained net.

Figure: Visible layer initialized to 10 examples from triangle dataset. Sampled after every 2 Gibbs cycles on **Diss-CD** trained net.

Outline

Diss-CD

Anomaly Detection

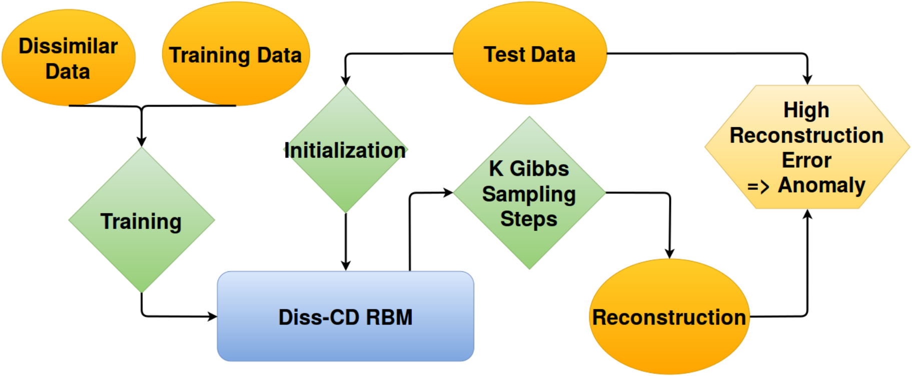


Figure: Anomaly Detection Pipeline

Diss-CD

Results

Reconstruction Errors

	PCD	Diss-CD
Silhouttes	367.7	**465.6**
CIFAR10	199.7	**251.2**
Triangles	85.7	**113.6**
MNIST	**56.6**	48.1

Table: Reconstruction Error comparison between PCD and Diss-CD trained RBMs on different datasets (Higher is better for anomaly detection). Only for MNIST, Diss-CD gives a lower reconstruction error!

Conclusions

1. Diss-CD, a semi-supervised method of training RBMs, shows promise for applications in anomaly detection.
2. Performance of Diss-CD depends on the choice of dissimilar data.
3. Diss-CD gives accurate reconstructions of training data when sampling from the net.

Diss-CD

Next Step?

- RBM hard to train for gaussian stochastic units
- Unable to achieve convergence.
- Return with more insight

Outline

Bonnet's Theorem

Let $f(x)$: $\mathbb{R}^d \to \mathbb{R}$ be a integrable and twice differentiable function. The gradient of the expectation of $f(x)$ under a Gaussian distribution $\mathcal{N}(x|\mu, \Sigma)$ with respect to the mean μ can be expressed as the expectation of the gradient of $f(x)$.

$$\nabla_{\mu_i} \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[f(x)] = \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[\nabla_{x_i} f(x)]$$

Price's Theorem

Let $f(x)$: $\mathbb{R}^d \rightarrow \mathbb{R}$ be a integrable and twice differentiable function. The gradient of the expectation of $f(x)$ under a Gaussian distribution $\mathcal{N}(x|\mu, \Sigma)$ with respect to the covariance Σ can be expressed in terms of the expectation of the Hessian of $f(x)$ as:

$$\nabla_{\Sigma_{i,j}} \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[f(x)] = \frac{1}{2} \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[\nabla^2_{x_i,x_j} f(x)]$$

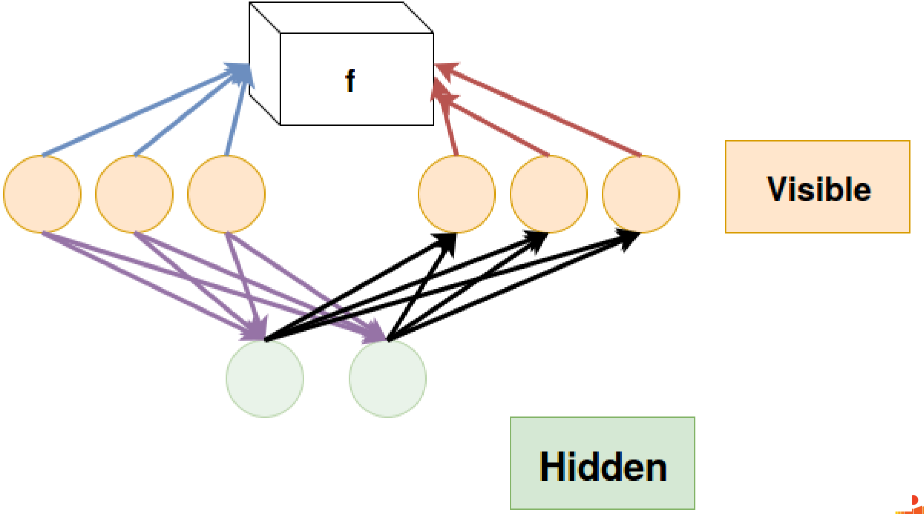
Backpropagation

By applying chain rule, we get:

$$\nabla_\theta \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[f(x)] = \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}\left[\mathbf{g}^T \frac{\partial \mu}{\theta} + \frac{1}{2} Tr\left(H \frac{\partial \Sigma}{\partial \theta}\right)\right]$$

$\mathbf{g}$: gradient
H: hessian

Relation to RBM



GB-RBM

Energy Function

$$E(v,h) = \sum_{i=1}^{V} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \boldsymbol{c}^T\boldsymbol{h} - \sum_{j=1}^{V}\sum_{i=1}^{H} \frac{v_i}{\sigma_i} h_j w_{ij}$$

Probability

$$P(v) = \sum_{\boldsymbol{h}} \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \sum_{\boldsymbol{h}} \frac{1}{Z} e^{-\sum_{i=1}^{V} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \boldsymbol{c}^T\boldsymbol{h} + \sum_{j=1}^{V}\sum_{i=1}^{H} \frac{v_i}{\sigma_i} h_j w_{ij}}$$

$$P(v) = \frac{1}{Z} e^{-F(v)}$$

where $F(v)$ is free energy

GB-RBM

Free Energy

$$F(v) = -log\Big(\sum_{\boldsymbol{h}} e^{-\sum_{i=1}^{V}\frac{(v_i-b_i)^2}{2\sigma_i^2}+\boldsymbol{c_h}^T\boldsymbol{h}+\sum_{j=1}^{V}\sum_{i=1}^{H}\frac{v_i}{\sigma_i}h_j w_{ij}}\Big)$$

Simplifying the term within the *log*

$$\sum_{\boldsymbol{h}} e^{-\sum_{i=1}^{V}\frac{(v_i-b_i)^2}{2\sigma_i^2}+\boldsymbol{c}^T\boldsymbol{h}+\sum_{j=1}^{H}\sum_{i=1}^{V}\frac{v_i}{\sigma_i}h_j w_{ij}}$$

$$= e^{-\sum_{i=1}^{V}\frac{(v_i-b_i)^2}{2\sigma_i^2}} \times \prod_j \big(e^{c_j+\sum_{i=1}^{V}\frac{v_i}{\sigma_i}w_{ij}} + 1\big)$$

Substituting, we get

$$F(v) = \sum_{i=1}^{V}\frac{(v_i-b_i)^2}{2\sigma_i^2} - \sum_j log\big(e^{c_i+\sum_{i=1}^{V}\frac{v_i}{\sigma_i}w_{ij}} + 1\big)$$

Outline

Big Picture

Objectives

To generate adversarial examples, a "measure" of dissimilarity is required. Instead of learning a distance metric, we wish to learn a "divergence". [2]

Bigger Picture

For Now

- To explore machine learning from a bayesian perspective
- Find methods to learn probability distribution over well-behaved manifolds

References I

Daan Wierstra Danilo Jimenez Rezende Shakir Mohamed. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *ICML* (2014). DOI: arXiv:1401.4082[stat.ML]. URL: https://arxiv.org/abs/1401.4082.

Max Welling Diederik P Kingma. "Auto-Encoding Variational Bayes". In: *The International Conference on Learning Representations (ICLR)* (2014). DOI: arXiv:1312.6114[stat.ML]. URL: https://arxiv.org/abs/1312.6114.

Technobium.com. URL: http://technobium.com/wordpress/wp-content/uploads/2015/04/MultiLayerNeuralNetwork.png.

References II

Wikipedia. URL: https://en.wikipedia.org/wiki/File:Boltzmannexamplev2.png.

Wikipedia. URL: https://en.wikipedia.org/wiki/File:Restricted_Boltzmann_machine.svg.